# Software Size Measurement of Knowledge Management Portal with Use Case Point

Karto Iskandar[1], Ford Lumban Gaol[2], Benfano Soewito[3], Harco Leslie Hendric Spits Warnars[4], Raymondus Kosala[5]
[1,5] School of Computer Science, [1,2,4] Doctor of Computer Science, [3] Binus Graduate Program
Bina Nusantara University
Jakarta, Indonesia
[1] karto_i@binus.edu, [2] fgaol@binus.edu, [3] bsoewito@binus.edu, [4] shendric@binus.edu, [5] rkosala@binus.edu

***Abstract*—Knowledge Management portal is a system to support Knowledge Management process, in order to create, capture, develop, share, reuse and optimize the knowledge and particularly in Bina Nusantara University which has implemented Knowledge Management System (KMS) since 2002. However, this KMS need to be measured in order to know how better this KMS in term of the software size. The BINUS KMS will be measured in term of their software size in functionality perspective with use case point method. This metric of KMS will be used by management to know how better the software size, complexity level and effort to development in numbering. Measurement of software size with software metric such as Use Case Point upon use case diagram for BINUS knowledge Management Portal shows that the project has medium software size with score Use Case Point (UCP) = 108.56 and has estimate effort will be developed in 2,064 hours (or in 258 days or 51.6 weeks or 12.9 months) and has development cost for 516,000,000.00 rupiah (Indonesian currency). Use Case Point, estimate effort and project value will powerful to help management in order to make decision regarding the implementation of IT software project development in term of time, money and people.**

***Keywords—Software Measurement; Software Metrics; Use Case Point; Knowledge Management Portal; BINUS University***

## I. INTRODUCTION

Knowledge Management (KM) is the process of creating, capturing, developing, sharing, and optimizing the organizational knowledge. Knowledge Management System (KMS) consist of people, process and technology which will collaborate in order to capture, maintain and deliver the knowledge's. Knowledge management System includes a lot of approaches to collect information, then this system builds the knowledge that can be searched through specialized search tools and or visual search tools that present information to used and optimized with specific user purpose. [1].

Bina Nusantara University or known as BINUS University is one private university which located in Jakarta, Indonesia and in order to support the process of Knowledge Management, a web based Knowledge Management Portal (KMP) were built in 2002 with IBM Lotus note. This KMP is continue to be developed with adding several technology and in 2008 was replaced with Microsoft SharePoint 2013. This KMP was built with purposes to support BINUS University business process activities in order to create, capture, develop, share, reuse and optimize their knowledge's [2].

Some of university think that KMP is a big system, complicated, very high complexity and need effort to develop it [1],[2] and not all universities in Indonesia have a Knowledge Management Portal (KMP) implementation to support their Knowledge Management process. BINUS University has implemented KMP since 2002 and still growing to complete the features until today. However, we need to know how better this BINUS KMP by measuring of the KMP software size, where software size can be measured based on length, functionality, complexity and reuse. This BINUS KMP software size will be measured in functionality perspective only with use case point approach.

Use case point is one of the approved software metric which can measure our software size based on functionality [3], [4]. Use-Case Points method can measure the software size, the complexity and effort how many man-days to develop this KMS [3], [5], [6]. In software development, use case is a tool which can be opted in user requirement, analyse and design steps in software development methodology [5], [6]. Thus, the purpose of this study is to measure BINUS University KMP using Use Case Point approach, than the result can be used by management to know the software size, complexity level and effort to development. Use case point has been implemented in project estimation for several application domain, such as: [7] Banking, CRM for bank, Finance, Internet shopping, Real-time system and much more. This method also has been implemented in a lot of company such as: [6], [7], Nageswaran: American Software Company, Mogel, IBM, Student Project, Agilis Solutions and FPT Software partnered, and much more.

In this paper, we will calculate the KMP at BINUS University software size using Use-Case Points method. a description and explanation about Use-Case Point will be presented in Section 2. In this section also telling previous related work with software measurement with Use Case Points. Section 3 will present in detail the original Use Case Points method that we use in KMP BINUS University measurement. In section 4, we briefly describe analysis the result of calculation software metric based on Use Case Points from previous section. In this section, we will analyze the KMP size, the complexity, and effort to develop BINUS University's KMP. Finally, in Section 5 we present the conclusions.

## II. USE CASE POINTS METHOD

In this section, we will describe the Use-Case Points method, step by step in calculation Use-Case Points and previous related work.

### A. Use-Case Points Method

International Function Point User Group (IFPUG) is non-profit organization which managed the evolution of the technique through several releases of the Function Point (FP) [5], [8], [9]. There are several variations of Function Point and methods with all pros and cons for each method, such as: Mark II, NESMA, FiSMA, COSMIC, FSM, FPA UML-Base and Use-Case Points [5], [7], [8]. The use case points method was proposed by Karner [7] and The method is extension of Function Points (FP) method [6], [9]. The detail step by step in Use-Case Points will describe in path B bellow.

Use-Case points (UCP) is a software size estimation calculation from use case model for software development project, where the model consists of the use case diagram document, context diagram, preconditions, post conditions, flow of events, subordinate use case diagrams, activity diagram, sequence diagrams, user interface, business rules, special requirements and other artifact documents related to software development [10], [11]. Use-Case points (UCP) method is related to functional, technical and environment complexity of the software project. The factors are number and complexity of the actors and use cases that contained in the use case diagram [5]. The complexity of actors and use cases in the system to quantify the variables Unadjusted Actor Weight (UAW) and Unadjusted Use Case Weight (UUCW), respectively. Later on, the size and complexity of the system that called Unadjusted called Use Case Points (UUCP) then calculated by combined between this UAW and UUCW multiply with their weight. The next step is to adjust this measure with a number of technical factors and environmental factors given by Technical Complexity factor (TCF) and Environmental Factor (EF) variables, respectively. These two factors combined with the UUCP to produce the effective number of adjusted Use Case Points (UCP). This number reflect the size and complexity of the software project [3], [5], [11].

### B. Step by step calculation Use-Case Points

To measure the software estimation using Use-Case Points approach following these 6 steps [5], [7], [11]:

1. Identify and weight Actors with Unadjusted Actor Weights (UAW)

   The actors in the use case model are categorized as simple, average and complex actors and have Weight Factor 1, 2 and 3 respectively. Actor in use case diagram can be categorized as simple actor when its actor with another system with defined Application Programming Interface (API). Meanwhile, average actor is another system that interact through protocol such as Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), and a data store either in files or Database Management System (DBMS) or a person who interact through text-based interface. Finally, complex author as a person who interact through Graphical User Interface (GUI). The number of UAW as shown in formula (1) is calculated by adding of calculation of number of actors and chosen weighted factor.

$$UAW = \sum (\#\ Actors * Weight\ Factor) \quad (1)$$

2. Identify and weight Use Cases with Unadjusted Use Case Weights (UUCW)

   Each use case is categorized as simple, average and complex, where simple use case is a use case with 3 or fewer transactions and has Weight Factor = 5. Meanwhile, average use case has 4 to 7 transactions in each use case and has Weight Factor = 10, and complex use case is a use case with more than 7 transactions and has Weight Factor = 15. Formula (2) shows that UUCW is a sum from multiplication between each of use case with Weight Factor.

$$UUCW = \sum (\#\ Use\ Cases * Weight\ Factor) \quad (2)$$

3. Calculate Unadjusted Use Case Points (UUCP)

   UUCP are counted based on adding two components, that is: the Unadjusted Actor Weight (UAW) and the Unadjusted Use Case Weight (UUCW) based on the total number of activities or use case as shown in formula (3).

$$UUCP = UAW + UUCW \quad (3)$$

4. Technical Complexity Factor (TCF) & Environmental Factor (EF)

   The software size does not depend only on its functions and users on the use case diagram, then UCP calculation need to assign values from the technical and environmental factors. There are 13 TCF which is marked with T1 till T13 as shown in Table III include its weight and they are : T1 for Distributed system, T2 for Response or throughput performance objectives, T3 for End-user efficiency, T4 for Complex internal processing, T5 for Reusable code, T6 for Easy to install, T7 for Easy to use, T8 for Portable, T9 for Easy to change, T10 for Concurrent, T11 for Includes security features, T12 for Provides access for third parties, T13 for Special user training facilities are required. Each of T1 till T13 weight in Table III are multiplied with score value and be sum into TCF as shown in formula (4).

   Similarly, There are 8 EF which is marked with F1 till F8 as shown in Table IV include its weight and they are: F1 for Familiar with Rational Unified Process, F2 for Application experience, F3 for Object-oriented experience, F4 for Lead analyst capability, F5 for Motivation, F6 for Stable requirements, F7 for Part-time workers, F8 for Difficult programming language.

Each of F1 till F8 weight in Table IV are multiplied with score value and be sum into EF as shown in formula (5).

$$TCF = 0.6 + (0.01 * TFactor) \quad (4)$$

$$EF = 1.4 + (-0.03 * EFactor) \quad (5)$$

5. Calculate (Adjusted) Use Case Points (UCP)

   Subsequently, based on above variables present in the previous step, the next calculation is to determine the effective number of UCP of the system using formula (6) where UUCP from formula (3) will be multiplied with TCF and EF in formula (4) and (5) respectively.

$$UCP = UUCP * TCF * EF \quad (6)$$

6. Estimate effort (E) in person-hours.

   After all, the last step is to calculate the effective effort (E) in person-hours (PH) by multiplying the specific value of Person Hour per UCP (PHperUCP) with the UCP from formula (6).

$$E = UCP * PHperUCP \quad (7)$$

### C. Previous Related Work

Many related work to measures method have been proposed for measure the size of the functionality software systems. Some measures performed by experts and it is performed using a formal estimation method like Use-Case Point. Anda et al. [7] reported that the application of Use-Case Points is affected by the several aspects of the structure of the use-case diagram, which use the generalization actors and (included and extended) use cases. Also, Anda et al. [7] presented the software size measurement based on use case points experience. Meanwhile, Diev [12] defined a heuristic application for Use-Case Points and rules in UML models in order to make the Function Point counting possible.

Moreover, Yavari et al. [13] using the Use Case point method and answer its challenge about how to determine unadjusted Use Case weight. This study find that Use Cases complexity metric is one of the most causal of inaccuracy and variation in estimation based on Use Case points considered. This study focused on Use Cases and other resources related to Use Cases, and identify other metrics to determine Use Case complexity and then calculating unadjusted Use Case weight. Furthermore, Levesque et al. [14] on their study aimed to automating the software measurement with function points approached. The advantage of this method is from the UML standard modeling notation and the training done with software engineers to model the software with the UML notation. Instead of using one of the function point methods that is difficult to apply and subject to too much interpretations, the software development only concentrate on software modeling using UML in developing application.

## III. BINUS UNIVERSITY KMP CASE STUDY

In order to measure the Knowledge Management Portal (KMP) software at BINUS University using Use-Case Points to get the software size, the complexity and effort to develop, then the BINUS KMP use case diagram are drawn based on previous research as can be seen in Fig. 1 [2]. Fig. 1 presents 3 Actors and 11 use cases where consist of 8 use cases and 3 include use cases.

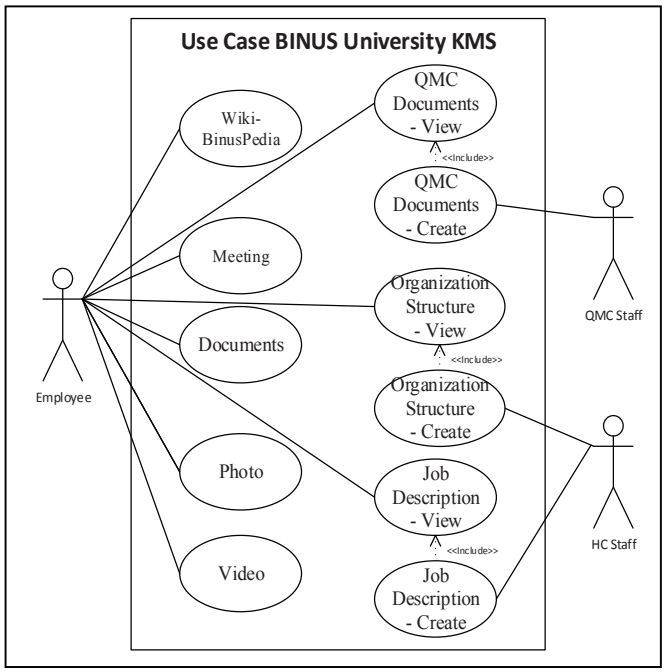


Fig. 1. Use Case Diagram BINUS University KMS

The use case diagram in Fig. 1 consist of 11 use case activities such as:

1. Use case Wiki-BinusPedia, this use case tell about general knowledge or information about BINUS University.
2. Use case Meeting, this use case tell about how to collaboration team meeting, sharing file, discuss a forum, and store the minute of meeting.
3. Use case Documents, this use case tell about all BINUS documents repository management.
4. Use case Photo, this use case tell about all BINUS photo collection management.
5. Use case Video, this use case tell about all BINUS video collection management.
6. Use case QMC Documents View, this use case tell about view or read QMC Documents. QMC is Quality Management Center, this center manage all quality standard at BINUS University.
7. Use case QMC Documents Create, this use case tell about create QMC Documents by QMC staff.

8. Use case Organization Structure View, this use case tell about view or read Organization Structure documents.
9. Use case Organization Structure Create, this use case tell about create Organization Structure by HC staff.
10. Use case Job Description Views, this use case tell about view or read Job Description documents.
11. Use case Job Description Create, this use case tell about create Job Description by HC staff.

The first step in Use-Case Points is to identify and weight Actors (UAW) in Fig. 1. The actors in Fig. 1 are categorized as complex actor since all the actors are person who interact with GUI and each has weight factor 3. As shown in Table I, the number of UAW = 9 as result from formula (1) where sum from multiplication of each weight score with number of actors.

TABLE I. IDENTIFY AND WEIGHT ACTORS (UAW)

| Category | Weight | Actors | Count | Weight * Count |
|---|---|---|---|---|
| Simple | 1 | - | 0 | 0 |
| Average | 2 | - | 0 | 0 |
| Complex | 3 | Employee, HC Staff, QMC Staff | 3 | 9 |
| Unadjusted Actor Weight (UAW) | | | | 9 |

The second step in Use-Case Points is to identify and weight Use Cases (UUCW) in Fig. 1. The 11 use cases in Fig. 1 are categorized as simple, average and complex. There are 5 simple use cases such as Wiki-BinusPedia, Documents, QMC Documents View (include), Organization Structure View (include) and Job Description Views (include)). Meanwhile, there are 4 average use cases such as Photo, Video, Organization Structure Create and Job Description Create and 2 complex use cases such as Meeting and QMC Documents Create. Table I shows the categorization of use cases in Fig. 1, and each categorization such as simple, average and complex are weighted with 5, 10 and 15 respectively. The number of UUCW = 95 as result from formula (2) where sum from multiplication of each weight score with category of each use case.

The third step in Use-Case Points is calculate Unadjusted Use Case Points (UUCP) by adding the total weight for Unadjusted Actor Weight (UAW) to the total for Unadjusted Use Case Weight (UUCW) based on formula (3) and the number of UUCP = 104 as shown below on equation (8). The number of UAW = 9 as shown in Table I while number of UUCW = 95 from Table II.

$$UUCP = 9 + 95 = 104 \quad (8)$$

TABLE II. IDENTIFY AND WEIGHT USE CASES (UUCW)

| Use Case | Category | Weight |
|---|---|---|
| Wiki-BinusPedia | Simple | 5 |
| Meeting | Complex | 15 |
| Documents | Simple | 5 |
| Photo | Average | 10 |
| Video | Average | 10 |
| QMC Documents View (include) | Simple | 5 |
| QMC Documents Create | Complex | 15 |
| Organization Structure View (include) | Simple | 5 |
| Organization Structure Create | Average | 10 |
| Job Description Views (include) | Simple | 5 |
| Job Description Create | Average | 10 |
| Unadjusted Use Case Weight (UUCW) | | 95 |

The fourth step in Use-Case Points is calculate Technical Complexity Factor (TCF) from formula (4) and Environmental Factor (EF) from formula (5) as shown in Table III and Table IV respectively. The number of TCF is 0.9 as shown below on equation (9), where TFactor is 30 as sum of factors in Table III, while EF is 1.16 as shown below on equation (10) with EFactor is 8 as sum of factors in Table IV.

$$TCF = 0.6 + (0.01 * 30) = 0.9 \quad (9)$$

$$EF = 1.4 + (-0.03 * 8) = 1.16 \quad (10)$$

The fifth step is calculate (Adjusted) Use Case Points (UCP) by using formula (6) where UUCP from formula (3) is multiplied with TCF and EF from formula (4) and (5) respectively. The number of UCP is 108.576 as shown below on equation (11) and rounded to 108.58.

$$UCP = 104 * 0.9 * 1.16 = 108.576 \quad (11)$$

The sixth step is calculate Estimate effort (E) in person-hours, by multiplying the UCP from formula (6) with defined PHperUCP as number of person hour per Use Case Point. Based on interview with BINUS IT department, this projects is categorized as simple/low category as shown in Table V [15], where each project needs 1 to 20 person hour, and PHperUCP for this project is 19. Based on formula (7) then UCP = 108.58 is multiplied with 19 and E as Estimation Effort is 2,063.02 as shown below on equation (12) and rounded to 2,064 hours.

$$E = 108.58 * 19 = 2{,}063.02 \quad (12)$$

TABLE III. CALCULATE TECHNICAL COMPLEXITY FACTOR (TCF)

| Factor | Weight | Value | Weight * Value |
|---|---|---|---|
| T1 Distributed system | 2 | 0 | 0 |
| T2 Response or throughput performance objectives | 2 | 0 | 0 |
| T3 End-user efficiency | 1 | 5 | 5 |
| T4 Complex internal processing | 1 | 1 | 1 |
| T5 Reusable code | 1 | 3 | 3 |
| T6 Easy to install | 0.5 | 4 | 2 |
| T7 Easy to use | 0.5 | 4 | 2 |
| T8 Portable | 2 | 1 | 2 |
| T9 Easy to change | 1 | 4 | 4 |
| T10 Concurrent | 1 | 3 | 3 |
| T11 Includes security features | 1 | 3 | 3 |
| T12 Provides access for third parties | 1 | 0 | 0 |
| T13 Special user training facilities are required | 1 | 5 | 5 |
| **TFactor** | | | **30** |

TABLE IV. CALCULATE ENVIRONMENTAL FACTOR (EF)

| Factor | Weight | Value | Weight * Value |
|---|---|---|---|
| F1 Familiar with Rational Unified Process | 1.5 | 2 | 3 |
| F2 Application experience | 0.5 | 2 | 1 |
| F3 Object-oriented experience | 1 | 3 | 3 |
| F4 Lead analyst capability | 0.5 | 4 | 2 |
| F5 Motivation | 1 | 3 | 3 |
| F6 Stable requirements | 2 | 3 | 6 |
| F7 Part-time workers | -1 | 5 | -5 |
| F8 Difficult programming language | -1 | 5 | -5 |
| **EFactor** | | | **8** |

TABLE V. SOFTWARE COMPLEXITY PHPERUCP CATEGORIES ON BINUS IT DEPARTMENT

| Category | Person-Hours per UCP (PHperUCP) |
|---|---|
| Simple / Low | 1 – 20 |
| Complex | 21 – 40 |
| Very complex / High | > 41 |

## IV. RESULT AND ANALYSIS

In this section, we will analyze the result of calculation software metric based on Use Case Points from previous section. Based on interview in BINUS IT department, there are 4 categories of software size by Use Case Points as shown in Table VI [15], they are Small, Medium, and Large with range number of Use Case Points around less equal than 99, 100 to 299 and greater equal than 300 respectively. Based on result of formula (6), the number of UCP as Use Case Points for this BINUS knowledge Management Portal is 108.58 and based on software size categories in Table VI, software size for BINUS knowledge Management Portal is categorized as Medium.

TABLE VI. SOFTWARE SIZE CATEGORIES ON BINUS IT DEPARTMENT

| Category | Use Case Points (UCP) |
|---|---|
| Small | <= 99 |
| Medium | 100 - 299 |
| Large | > = 300 |

Meanwhile, based on interview with BINUS IT department and BINUS University documentation regulation [15], BINUS University has 5 days working with 8 hours per day. Based on result of formula (7) where E as estimate effort for this BINUS knowledge Management Portal is 2,064 hours, on other words this BINUS knowledge Management Portal should be developed maximum in 2,064 hours. Based on BINUS University working day-hour with 5 days and 8 hours working day then this estimate effort is divided with 8 and has 258 days (2,064/8=258). Moreover, this 258 days is divided with 5 as 5 days working and has 51.6 weeks (258/5=51.6). Furthermore, this 51.6 weeks is divided with 4 where there are 4 weeks in a month and has 12.9 months (51.6/4=12.9). Thus, this BINUS knowledge Management Portal should be developed maximum in 12.9 months. This project BINUS knowledge Management Portal can be shorted in development time by including many different IT staff roles which is scheduled with time schedule management such as Gantt chart.

Since this project BINUS knowledge Management Portal will include many IT roles with different salary per month based on their skill and expertise, then based on interview with BINUS IT department and BINUS university documentation regulation [15], the average for man-day is Rp. 2,000,000.00. Finally, this project BINUS knowledge Management Portal has project value around Rp. 516,000,000.00 where man-day Rp. 2,000,000.00 is multiplied with 258 as 258 days for developing this project (258 days * Rp. 2,000,000 = Rp. 516,000,000.00).

TABLE VII. SUMMARY OF THE BINUS KMP MEASUREMENT USING USE –CASE POINT

| Description | BINUS KMP |
|---|---|
| Unadjusted Actor Weight (UAW) | 9 |
| Unadjusted Use Case Weight (UUCW) | 95 |
| Unadjusted Use Case Points (UUCP) | 104 |
| Technical Complexity Factor (TCF) | 0.9 |
| Environmental Factor (EF) | 1.16 |
| (Adjusted) Use Case Points (UCP) | 108.58 |
| Software Size | 100 - 299 (Medium) |
| Software Complexity PHperUCP | 1-20 (19) (Simple) |
| Estimate effort (E) | 2,064 hours |
| Project Value | Rp. 516,000,000.00 |

## V. CONCLUSIONS

Measurement of software size with software metric such as Use Case Point is measurement based on internal product attribute measurement which can be measured only based on the entity itself. Moreover, Use Case Point is part of measurement software size in term of functionality where function supplied is measured by the product to the user. Measurement with Software size can reflect effort, cost and productivity.

Measurement of software size with Use Case Point is useful where size of software can be measured based on use case diagram. In this paper use case diagram for project BINUS knowledge Management Portal is measured with Use Case Point and the measurement shows that this project has medium software size with score Use Case Point (UCP) = 108.56 and has estimate effort will be developed in 2,064 hours (or in 258 days or in 51.6 weeks or in 12.9 months) and has development cost for Rp. 516,000,000.00 as shown in Table VII.

Measurement of software size with Use Case Point can help management to make better decision in term of how to deal with the project, how long to develop and how much to invest. Moreover, based on this Use Case Point Score, estimate effort and project value, the number of people who involve in this project can be measured.

## REFERENCES


[1] E. Gourova and K. Toteva, "Design of Knowledge Management Systems," *Proc. 20th Eur. Conf. Pattern Lang. Programs*, pp. 1–15, 2015.

[2] K. Iskandar, "Pengembangan Knowledge Management System pada Bina Nusantara Berbasis Microsoft Sharepoint 2010," *ComTech*, vol. 5, no. 01, pp. 429–440, 2014.

[3] N. Fenton and J. Bieman, *Software Metrics A Rigorous and Practical Approach*, Third Edit. London: CRC Press, 2015.

[4] A. B. Nassif, L. F. Capretz, and D. Ho, "Calibrating use case points," *Companion Proc. 36th Int. Conf. Softw. Eng. - ICSE Companion 2014*, pp. 612–613, 2014.

[5] L. M. Alves, P. Ribeiro, and R. J. Machado, "An Empirical Study on the Estimation of Software Development Effort with Use Case Points," *IEEE Front. Educ. Conf.*, pp. 101 – 107, 2013.

[6] R. K. Clemmons, "Project Estimation With Use Case Points," *J. Def. Softw. Eng.*, pp. 18–22, 2006.

[7] B. Anda, H. C. Benestad, and S. E. Hove, "A Multiple-Case Study of Software Effort Estimation based on Use Case Points," *Int. Symp. Empir. Softw. Eng.*, pp. 407–416, 2005.

[8] L. Lavazza and G. Robiolo, "Introducing the Evaluation of Complexity in Functional Size Measurement: a UML-based Approach," *Proc. 2010 ACM-IEEE Int. Symp. Empir. Softw. Eng. Meas. - ESEM '10*, p. 1, 2010.

[9] G. B. Ibarra and P. Vilain, "Software Estimation Based on Use Case Size," *Softw. Eng. SBES 2010 Brazilian Symp.*, pp. 178–187, 2010.

[10] E. R. Carroll, "Estimating software based on use case points," *Companion to 20th Annu. ACM SIGPLAN Conf. Object-oriented Program. Syst. Lang. Appl. - OOPSLA '05*, pp. 257–265, 2005.

[11] S. Kusumoto, F. Matukawa, K. Inoue, S. Hanabusa, and Y. Maegawa, "Estimating effort by use case points: Method, tool and case study," *Proc. - Int. Softw. Metrics Symp.*, pp. 292–299, 2004.

[12] S. Diev, "Use Cases Modeling and Software Estimation: Applying Use Case Points," *ACM SIGSOFT Softw. Eng. Notes*, vol. 31, no. 6, pp. 1–4, 2006.

[13] Y. Yavari, M. Afsharchi, and M. Karami, "Software complexity level determination using software effort estimation use case points metrics," *2011 Malaysian Conf. Softw. Eng.*, no. 3, pp. 257–262, 2011.

[14] G. Levesque, V. Bevo, and D. T. Cao, "Estimating software size with UML models," *Proc. 2008 C3S2E Conf. - C3S2E '08*, pp. 81–87, 2008.

[15] IT_BINUS_Department, "BINUS IT Software Development Documentation," Jakarta, Indonesia, 2016.